

**Method and System for Automized and Synchronous Execution of Customized Code on
Off-loaded Retrieved Documents in a Document Processing System**

BACKGROUND OF THE INVENTION

Related Inventions

5 IBM patent application serial number _____, entitled "Technique for Content Off-Loading in a Document Processing System Using Stub Documents", filed concurrently herewith on December 12, 2001 and IBM application serial number _____, entitled " Method and System for Off-Loading and Retrieving Document Content in a Document Processing System", filed concurrently herewith on December 12, 2001, are related to this invention.

10 Field of the Invention

The invention relates to document processing environments with large document repositories. More specifically, the present invention relates to a technique for processing off-loaded and/or retrieved documents in a document processing system.

Description of the Related Art

15 A problem associated with the use of known client mailing applications, such as Lotus™ Notes™ or Microsoft™ Outlook™, is that they contain continuously growing document repositories. (Lotus and Notes are trademarks of Lotus Development Corporation and/or International Business Machines Corp., and Microsoft and Outlook are trademarks of the Microsoft Corp.) The repositories continuously grow due to the incoming and outgoing notes or emails,
20 which will be referred to hereafter as documents, which are commonly saved until deleted by a user. Such documents often including large attachments like text documents, graphics or even storage consuming digitized pictures.

In response to this problem, a Lotus Notes application uses a Lotus Domino™ database

from which a tool like IBM Content Manager CommonStore™ for Lotus Domino (CSLD) is used to move documents stored in that database to an archive physically located on a different device, such as a tape storage. CSLD thereupon permits access to the archived documents. Domino and CommonStore are trademarks of Lotus Development Corp. and/or International Business Machines Corp.

CSLD also allows to access documents that have been archived from any archive client application (e.g., scanning applications, CommonStore for SAP™, etc.). When documents are retrieved from the archive to a Notes database, a Lotus Notes document is created. SAP is a trademark of SAP AG.

Lotus Notes supports writing of customized code which, for instance, can be used to trigger workflow based on the state of off-loaded documents.

The drawback of the existing approach is that customized code can not be invoked synchronous to an archiving/off-loading and retrieval process.

SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to provide a technique for enabling invocation of customized code synchronous to off-loading of content from a document in a document processing system to an archive.

Another object of the present invention is to provide a technique for enabling invocation of customized code synchronous to retrieval of content to a document in a document processing system from an archive.

Another object of the present invention is to provide a technique for synchronous code invocation in a document archiving environment which is transparent to an end user without requiring any user interaction.

The underlying idea is to provide a code component, in particular a plug-in, that is automatically started before a document is off-loaded, after it has been off-loaded or after it has been retrieved from the repository or archive.

5 The invention enables advantageous pre-processing and/or post-processing of documents in the above described document processing environment. This is accomplished by invocation of at least one agent and execution of customized code at a well-defined time, i.e., synchronous with the underlying document processing step or event.

BRIEF DESCRIPTION OF THE DRAWINGS

In the description that follows, the present invention is described in more detail by way of embodiments from which further features and advantages of the invention become evident, wherein:

Fig. 1 shows the various steps during archiving of a number of documents, invoking pre- and post-archiving agents in accordance with the invention; and

15 Fig. 2 shows an example of how pre-archiving and post-archiving agents can be applied to a document in a document processing system.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the flowchart of Fig. 1, a user of a document processing system, such as Lotus Notes, selects one or more documents for archiving (Step 101). Based on the archiving selection step, an archiving request is created (Step 102). In the present example, the request is made to IBM Content Manager CommonStore™ for Lotus Domino (CSLD). CSLD is a tool to move Lotus Notes documents in various formats to an archive.

CSLD then prepares one of the selected documents for archiving (Step 103). In accordance with the present invention, CSLD may invoke a pre-processing agent which executes code relative to the document currently being prepared for archiving (Step 104). When the agent is finished, CSLD archives the document (Step 105). Then, alternatively or in combination with Step 104, in accordance with the present invention, CSLD may invoke a post-processing agent on the current document (Step 106). When the agent is finished, CSLD marks the document as 'archived' (Step 107), and processing returns to Step 103 for processing the next selected document to be archived. Processing ends when all the selected documents have been processed.

Fig. 2 illustrates how the pre-archiving and post-archiving agents can be used to compute the amount of archived data and write it to a database for accounting purposes. Organizations often charge individuals or departments in the organization based on the amount of storage utilized, and the present invention can be used in this situation.

It is noted that the dotted lines in the diagram separate the main path of execution from the pre-processing and post-processing paths.

In Step 201, a user selects documents for archiving. An archiving request is then created in response to the selection step (Step 202). In the present example, this request is for the CSLD. CSLD then prepares the first document for archiving (Step 203) and invokes the pre-archiving agent on it. In Step 204, the code in this agent computes the current size of the document and the number of attachments in the document. These values are written to special fields in the current document (Step 205).

When the pre-archiving agent has finished, CSLD archives the document (Step 206) and invokes the post-archiving agent. The code in this agent removes attachments and rich text items from the current document (Step 207). The size of the modified document is then calculated (Step 208). The post-archiving agent then determines the size difference between the original document and the modified document, and logs the size difference in an accounting database (Step 209). Finally, when the post-archiving agent is finished, CSLD marks the document as 'archived' (Step 210). The processing is then repeated for the other selected documents.

A similar procedure can be implemented using post-retrieval agents. In such a scenario, when the detached portions of a document are reinserted thereto, such as when a user attempts to open an archived document, the amount of retrieved data is determined and written to an accounting database.

In CSLD, pre-processing and post-processing plug-ins are realized as the following agent invocation exits:

Pre-archiving agent: A Notes agent that is invoked on a document right before it is archived. This agent is usually used to prepare a document for archiving.

Post-archiving agent: A Notes agent that is invoked on a document after it has been archived successfully. This agent can be used to delete the document, move it to a certain folder, create a stub document to release resources, collect accounting information, trigger workflows, etc.

Post-retrieval agent: A Notes agent that is invoked on a document after it has been retrieved from an archive. This agent can be used to set security properties, set workflow flags, write the document to a folder, etc.

All automatic agents are configured based on the Notes document form, i.e., for documents of different forms, different agents can be configured. The agents are invoked synchronously. The

current document being archived/retrieved is passed to the agent via the session's document context. This permits the agent to be coded in LotusScript or Java. Errors occurring during agent execution are written to the CSLD trace file.